Software Engineering Institute

Insider Threat Control: Using Universal Serial Bus (USB) Device Auditing to Detect Possible Data Exfiltration by Malicious Insiders

George J. Silowash
Todd B. Lewellen

January 2013

TECHNICAL NOTE
CMU/SEI-2013-TN-003

CERT® Program

http://www.sei.cmu.edu

Carnegie Mellon

Copyright 2012 Carnegie Mellon University

This material is based upon work funded and supported by Department of Homeland Security under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Department of Homeland Security or the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution except as restricted below.

Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

External use:* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

* These restrictions do not apply to U.S. government entities.

Carnegie Mellon® and CERT® are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM-0000084

Table of Contents

List of Figures

Acknowledgments

Special thanks to our sponsors at the U.S. Department of Homeland Security, National Cyber Security Division, Federal Network Security branch for supporting this work.

Abstract

Universal serial bus (USB) storage devices are useful for transferring information within an organization; however, they are a common threat vector through which data exfiltration can occur. Despite the threat, many organizations feel that the utility of USB storage devices outweighs the potential risks. Implementing controls to track the use of these devices is necessary if organizations wish to retain sufficient situational awareness and auditing capabilities to detect data theft incidents.

This report presents methods to audit USB device use within a Microsoft Windows environment. Using various tools—the Windows Task Scheduler, batch scripts, Trend Micro's OSSEC host-based intrusion-detection system (HIDS), and the Splunk log analysis engine—we explore means by which information technology (IT) professionals can centrally log and monitor USB device use on Microsoft Windows hosts within an organization. In addition, we discuss how the central collection of audit logs can aid in determining whether sensitive data may have been copied from a system by a malicious insider.

1 Introduction

Malicious insiders attempting to remove data from organizational systems may have various ways of doing so, such as by using email and cloud storage services. Some malicious insiders attempt to remove data by using removable universal serial bus (USB) media.

As discussed in a prior Software Engineering Institute (SEI) report, *Insider Threat Control: Understanding Data Loss Prevention (DLP) and Detection by Correlating Events from Multiple Sources*, the use of removable media presents unique problems to the enterprise since insiders can use such media to remove proprietary information from company systems [1]. Insiders may do this for legitimate reasons, such as to work on material at home, or they may do so for malicious reasons, such as to steal intellectual property.

Staff members of the CERT® Program, part of Carnegie Mellon University's Software Engineering Institute, have seen instances where removable media played a role in a malicious insider's attack. Given this and other considerations which we discuss later in this report, organizations must establish and implement effective methods and processes to prevent unauthorized use of removable media while still allowing users with a genuine business need to access and remove such media. In addition, organizations should establish sound methods to track critical electronic assets so that they may better protect them.

This report presents methods to audit USB device usage within a Microsoft Windows environment. Using various tools—the Windows Task Scheduler, batch scripts, Trend Micro's OSSEC host-based intrusion-detection system (HIDS), and the Splunk log analysis engine—we explore means by which information technology (IT) professionals can centrally log and monitor USB device usage on Microsoft Windows hosts within an organization. In addition, we discuss how the central collection of audit logs can aid in determining whether sensitive data may have been copied from a system by a malicious insider. Implementing controls to track the usage of these devices is necessary if organizations wish to retain situational awareness and auditing capabilities during a data theft incident.

The methods described in this report are designed so that each Windows host will check for changes to its `USBSTOR` registry key every five minutes. Whenever a change is detected, due to either a new USB device being inserted or a previous one being re-inserted, the host will locally log the new registry values as well as the host's user-session information. At the same time, the host will (optionally) send a short SYSLOG message to a central log server for immediate alerting purposes. Additionally, the OSSEC HIDS system will centrally log the new registry values and session information and forward them to a Splunk system for analysis.[1] (We outline this process fully in Section 2.4.)

® CERT is a registered trademark owned by Carnegie Mellon University.

[1] We discuss OSSEC further in Section 2.5, and we present background about and uses for Splunk in Section 5.

This approach offers several key advantages that can assist an organization in its efforts to monitor potential incidents of data theft:

1. USB device usage can be detected quickly. It takes less than five minutes for the system to generate an alert.
2. There is redundant logging of information. Logs are stored locally on the hosts, centrally on an OSSEC server, and centrally on a SYSLOG server.
3. The system provides attribution. Current user-session information is logged when an incident is detected.
4. Native and open source tools are utilized. Local logging utilizes only Windows native capabilities and a single, open source, forensic executable; centralized logging can be done with the open source OSSEC system.
5. The system is customizable. An organization can choose to exclude any of the centralized logging capabilities and retain just the local logging capabilities or further modify the control to best suit its needs.

1.1 Audience and Structure of This Report

This report is a hands-on guide for system administrators and information security teams who are implementing USB device auditing and want to have a better understanding of which devices may be in use throughout the organization. We assume that readers are comfortable installing software and have a basic knowledge of how to edit a script.

The remainder of this report is organized as follows:

- Section 2 describes methods to establish proper auditing policies and technical controls to help reduce the risk of malicious insider activity.
- Section 3 presents additional information about USB audit logs.
- Section 4 lists benefits for 1) federal government agencies that use air-gapped systems and 2) organizations that want to protect intellectual property.
- Section 5 outlines how Splunk can be used to help identify events that are related to USB device usage.
- Section 6 summarizes this report.

2 Mitigating Insider Threat: Tools and Techniques

We define a *malicious insider* as a current or former employee, contractor, or business partner who

- has or had authorized access to an organization’s network, system, or data, and
- intentionally exceeded or misused that access, and
- negatively affected the confidentiality, integrity, or availability of the organization’s information or information systems

Malicious insiders are able to act within an organization by taking advantage of weaknesses they find in systems or by exploiting existing legitimate processes to their advantage. Organizations must be aware of such weaknesses and how an insider may exploit them; organizations must also be aware of the many ways in which weaknesses are introduced. For example, an organization may have insecure configurations or relaxed or nonexistent security policies. In other cases, a lack of situational awareness introduces weaknesses that malicious insiders can exploit, such as understanding how organizational policies effect employees ability to perform their job effectively and efficiently. Additionally, an organization that allows its employees to use USB storage devices is essentially increasing the potential for data leakage. Establishing proper auditing policies and technical controls, as discussed in this report, will mitigate some of the risks.

Our research has revealed that most malicious insider crimes fit into one of three categories: IT sabotage, theft of intellectual property, and fraud. This report focuses on the theft of intellectual property using removable media, in particular, USB devices. When USB devices are introduced into a Microsoft Windows-based system, the system generates numerous artifacts that can be audited or possibly used for forensic analysis. Therefore, it is important to understand how USB devices interact with the system.

The tools and techniques presented in the next sections represent just a subset of various practices an organization could implement to detect and mitigate insider threats. For example, organizations may wish to deploy commercially available software to prevent data loss. These tools and methods can be used by organizations of any size, and we intentionally selected open source and public domain tools since they are freely available to the public. Additionally, many of the commands that we present are native to the Windows operating system.

2.1 Utilizing the CERT Insider Threat Database

The CERT Program’s insider threat research is based on an extensive set of insider threat cases that are available from public sources, court documents, and interviews with law enforcement and/or convicted insiders, where possible. The database contains more than 700 cases of actual malicious insider crimes. Each case is entered into the database in a consistent, repeatable manner that allows us to run queries to search for specific information. The database breaks down the complex act of the crime into hundreds of descriptors, which can be further queried to provide statistical validation of our hypotheses. Since the database has captured very granular information about insider threat cases, it provides a way to find patterns of insider activity, discover possible precursors to insider attacks, and discover technical and nontechnical indicators of insider crime.

This helps us to establish trends and commonalities and identify techniques that may be helpful in mitigating insider threats.

2.2 Auditing

To effectively detect security incidents, organizations must develop auditing policies and procedures that reflect the organization’s needs. In particular, the policies should outline auditing requirements that may be defined by laws and other regulations. Auditing records and other logs will also aid in troubleshooting, help organizations hold users accountable for their actions, and assist in detecting and responding to security incidents.

To simplify log management, we recommend that logs be centrally collected and stored. Organizations can use a Security Information and Event Management (SIEM) solution to simplify log management. SIEM solutions allow information systems personnel to quickly search for events and automate alerts based on the organization’s needs. Depending on the organizational needs, SIEM solutions can require a significant investment; however, the payoff is great since such approaches help systems and information security teams focus on important events.

Smaller organizations may not be able to make an investment in a SIEM solution, or they may not be equipped to dedicate resources to establishing or maintaining such an approach. Such organizations may benefit from employing other solutions, such as using a centralized log server, which can collect events from various computers and network devices within the organization. Windows Vista and Windows 7 machines have the ability to have their logs centrally collected by Microsoft Windows Server 2008 R2.[2] Network devices typically send events to log servers using the SYSLOG protocol. Open source and commercial SYSLOG server packages are available to host a Microsoft Windows server-based SYSLOG system. Additionally, many versions of Linux, such as Ubuntu and Fedora, support SYSLOG natively and are open source.

This report focuses on using a Splunk instance on Microsoft Windows Server 2008 R2 for managing event logs.[3] Splunk has both a free and an enterprise version; one of the key differences in the free versus the enterprise version is the amount of logs that can be processed in one day. Currently, the free version caps log processing at 500MB per day. Depending on the organization’s needs, the free version may help smaller organizations get started in logging. The enterprise version offers additional features and is capable of processing larger volumes of data [2]. We chose to focus on Splunk since a free version of it is available and it can quickly search large quantities of data. We discuss Splunk in more detail in Section 5.

2.3 Control USB Removable Media

There are valid reasons for using USB removable storage devices within an organization. Removable media such as USB flash drives offer greater storage capacity and become less expensive each year; this has increased their popularity and use. However, organizations should consider issuing company-owned removable devices only to those employees who have a genuine business

[2] The Microsoft website contains information about centrally collecting logs [9]. Also, WindowSecurity.com offers a brief tutorial about centrally collecting logs [17].

[3] Splunk is a software package that indexes data from any source [2].

need for such devices. Company-owned devices allow an organization to better audit device usage, as described in Section 3. Additionally, organizations that issue devices can control the amount of storage these devices have, thereby limiting the amount of data that can be removed from the system at one time; such actions increase the chances of a malicious insider being caught as the organization reviews the audit logs. The organization should also establish policies and procedures that govern proper USB use. Many manufacturers augment the circuitry of their devices to “burn in” a serial number. Organizations can take advantage of this: by standardizing on a particular, make, model, and capacity, an organization can more easily track and audit these devices.

By centrally storing and auditing USB device events generated from the script (see Section 2.4) and combining that practice with using OSSEC (see Section 2.5), organizations can see where devices are used and better understand where problems may exist.

2.4 Conduct Windows USB Device Auditing with Scripts

As discussed in a prior SEI report titled *Insider Threat Control: Understanding Data Loss Prevention (DLP) and Detection by Correlating Events from Multiple Sources*, Windows-based systems have a built-in method to audit USB events based on changes to the registry [1]. However, without sifting through registry keys, it is difficult to use this method to quickly ascertain which USB devices were used on a system.

The CERT Insider Threat Center has developed a script, `usbHistory.bat`, that utilizes native Windows command line tools and a public domain tool known as USB History Dump. Section 2.4.1 describes the `usbHistory.bat` script. This script captures USB removable media usage information and enables an open source tool, known as OSSEC, to monitor changes to a system and alert on these changes.

2.4.1 The CERT Insider Threat Center Script

The `usbHistory.bat` script utilizes native Windows command line tools and a public domain tool known as USB History Dump. You must download and extract USB History Dump to the same path as the scripts.[4] The goal of the script is twofold. First, it gives organizations of any size the ability to generate an alert when a USB removable storage device is inserted into a monitored workstation; second, the script is able to generate this alert without requiring the organization to install an agent. To enhance these capabilities and make auditing logs more useful, we recommend using OSSEC, which utilizes an agent-based system for log collection. We discuss OSSEC further in Section 2.5, and we have included the Insider Threat Center script in the appendix of this report.

2.4.1.1 Prerequisites

Before implementing the `usbHistory.bat` script, you should determine where you will place the script and supporting files, configure the proper security settings for the files and folders, and configure alerts using Log Parser. Part of this process includes taking an important security meas-

[4] The complete source code and the executable file for USB History Dump are available from SourceForge [12]. In a document titled *USB History*, Nabiy, the author of the tool, describes how USB History Dump works [15].

ure to prevent end users from accessing the script directory. We provide more details about these steps in the next sections.

Determine the Location of the Script and Supporting Files

Before you can implement the auditing script, you must first decide how you will deploy the script and supporting files to each machine that will be audited. Various methods exist for deploying scripts. For example, you can use Group Policy and other system management software. However first, you will need to create a directory on each machine that will be audited. Staff members in the CERT Insider Threat Center installed the script and supporting files in the following directory:

```
C:\Admin_Tools\USB_Audit\
```

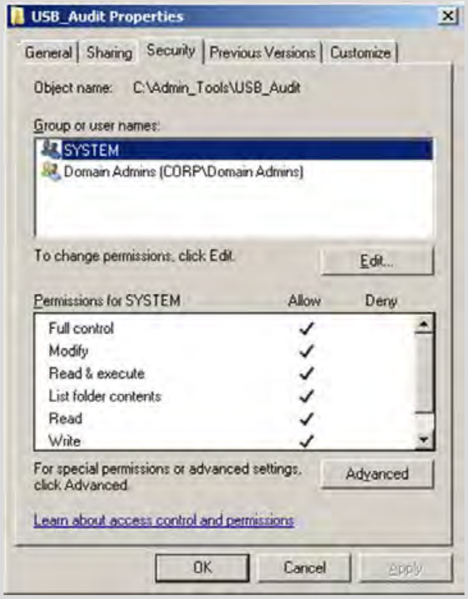
When selecting a deployment path, avoid using spaces in directory names since this will cause errors in the script.

Configure Proper Security Settings for File and Folder Permissions

You must configure proper security levels on the selected folder to prevent a regular user from writing to or modifying the script. If the proper settings are not configured, a malicious insider could modify the script so that the next time it executes, it could destroy a system. In the above case, we configured `USB_Audit` so that *only the following users have access*: SYSTEM and domain administrators. These users and groups should have full control of the directory. The script will execute under the SYSTEM context, and Domain Administrators must be able to access the script logs. All files contained within the script folder should inherit permissions from the parent folder.

Important Security Measure: Prevent End Users from Accessing the Script Directory

It is imperative that you prohibit end users from accessing the script directory. If a malicious user is able to modify the script, commands will execute with SYSTEM permissions since the scheduled task (defined later) requires SYSTEM permissions to function. See Figure 1 for an illustration of the proper file permissions to set on the script directory.



Object name: C:\Admin_Tools\USB_Audit

Group or user names:

- SYSTEM
- Domain Admins (CORP\Domain Admins)

To change permissions, click Edit.

Permissions for SYSTEM	Allow	Deny
Full control	✓	
Modify	✓	
Read & execute	✓	
List folder contents	✓	
Read	✓	
Write	✓	

For special permissions or advanced settings, click Advanced.

Figure 1: Proper Configuration of File Permissions for the Script Directory

Configure Alerts Using Microsoft’s Log Parser Tool

If you wish to send alerts to a central SYSLOG server, you will also need to use the *Log Parser* tool, which is available from Microsoft.[5] You must download and extract this tool to the same directory as the scripts.

Note that the Log Parser tool is part of an installer package. Because of this, an administrator should install this software on an administrative workstation or other suitable device and copy the tool’s executable (`logparser.exe`) to a directory that is included in the path environment variable.

CERT staff members tested Log Parser by placing the executable in the `C:\Windows\System32` directory, which is typically included in the system path by default. To see the currently defined system path of a machine, execute the following command from a command prompt:

```
echo %PATH%
```

Central SYSLOG logging is not required, but it will assist in quickly determining if a USB removable drive was ever connected to a system. Without central logging, an administrator will need to check individual logs on each workstation throughout the organization.

[5] Log Parser is available from Microsoft [17].

2.4.1.2 The USB Auditing Script

How the Script Works

The USB auditing script utilizes native Microsoft Windows command line utilities to ensure overall compatibility and to make it easier for administrators who are not familiar with other scripting languages, such as VBScript, to make additional customizations.[6] Combined with the built-in Windows commands, the script also uses a forensics tool, USB History Dump, to display USB removable devices that have been connected to the system.[7] This tool reads various values from the Windows Registry and typically displays them on the screen. However, when the tool is called from within the script, the output is directed to a file instead. This file is then compared to the last file that was generated. Detected differences are stored in a separate file, and if desired, the script generates a SYSLOG alert. OSSEC, discussed further in Section 2.5, utilizes the differences file and centrally stores it on its own server for further analysis and auditing.

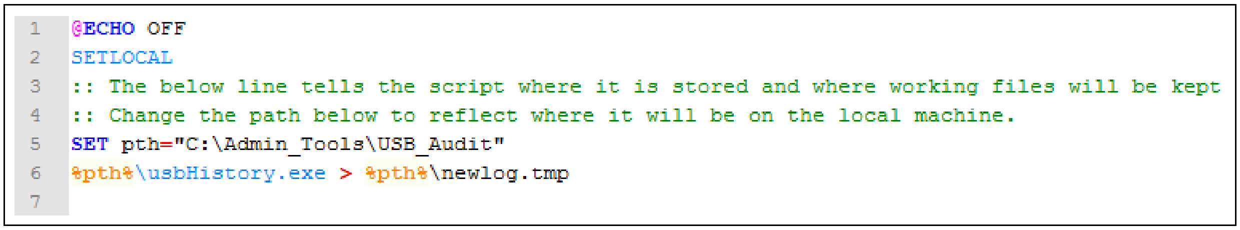
Modify the `usbHistory.bat` Script

You will need to modify the `usbHistory.bat` script to ensure that it reflects the correct configuration. The script has been commented to help administrators understand its instruction sequence. To change the script, simply open the file with any text editor.

[6] Other scripting languages may accomplish this task more eloquently. We encourage administrators to, when appropriate, investigate using scripts that best address the organization’s needs.

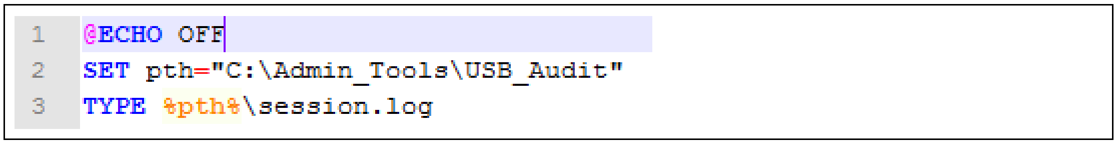
[7] See Section 2.4.1 for more information about USB History Dump.

Figure 2 shows a snippet of code from the `usbHistory.bat` script. On line 5, you will notice a local environment variable called `pth`. If you have stored the scripts and supporting files in a different location, you must change this variable. To do so, change the default of `C:\Admin_Tools\USB_Audit` to reflect the location of the script. This name should not contain spaces since this has shown to produce unreliable results during testing. You should also edit the `usbSession.bat` file to reflect the same path. Edit line 2 as shown in Figure 3.



```
1  @ECHO OFF
2  SETLOCAL
3  :: The below line tells the script where it is stored and where working files will be kept
4  :: Change the path below to reflect where it will be on the local machine.
5  SET pth="C:\Admin_Tools\USB_Audit"
6  %pth%\usbHistory.exe > %pth%\newlog.tmp
7
```

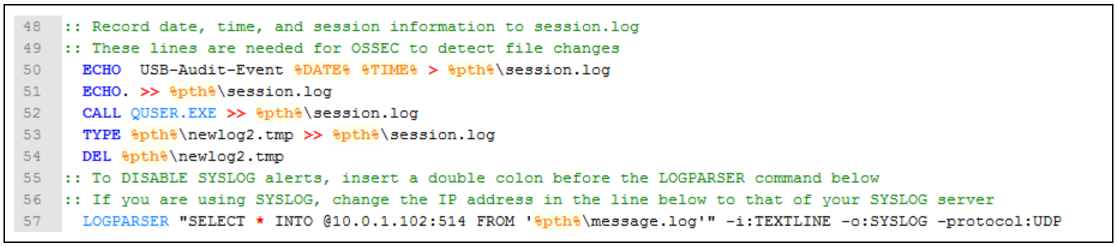
Figure 2: Set the `pth` Environment Variable



```
1  @ECHO OFF
2  SET pth="C:\Admin_Tools\USB_Audit"
3  TYPE %pth%\session.log
```

Figure 3: Provide the Correct Path in the `usbSession.bat` File

By default, the script is configured to send `SYSLOG` alerts to a central server; the `SYSLOG` server listed in the script is currently set to 10.0.1.102:514 as noted on line 57 of the script (see Figure 4). You should change the IP address to reflect the address of your organization's `SYSLOG` server.



```
48 :: Record date, time, and session information to session.log
49 :: These lines are needed for OSSEC to detect file changes
50   ECHO  USB-Audit-Event %DATE% %TIME% > %pth%\session.log
51   ECHO. >> %pth%\session.log
52   CALL QUSER.EXE >> %pth%\session.log
53   TYPE %pth%\newlog2.tmp >> %pth%\session.log
54   DEL %pth%\newlog2.tmp
55 :: To DISABLE SYSLOG alerts, insert a double colon before the LOGPARSER command below
56 :: If you are using SYSLOG, change the IP address in the line below to that of your SYSLOG server
57   LOGPARSER "SELECT * INTO @10.0.1.102:514 FROM '%pth%\message.log'" -i:TEXTLINE -o:SYSLOG -protocol:UDP
```

Figure 4: Configure SYSLOG to Send Alerts to a Central Server

Disable SYSLOG Alerts

If you wish to disable SYSLOG alerts, simply place a double colon (`::`) before the `LOGPARSER` command on line 57. Be sure that your central SYSLOG server is configured to receive alerts from a particular source IP address. You may need to adjust firewall rules, both on the server and any network firewalls, to permit the alerts as well. If you are using Splunk to receive and process logs, be sure to configure it to receive the SYSLOG events.

Execute the Script Regularly

Finally, you should configure the script to execute on a fairly regular basis. You should also configure a scheduled task to execute the script; you can use Group Policy to push a scheduled task to multiple machines.[8] Access the task scheduler by navigating to Start → All Programs → Accessories → System Tools → Task Scheduler. Configure a new task using the following parameters:

1. In the General tab
   a. Name the task `USB Device Audit`.
   b. Check the box next to *Run with highest privileges*.
   c. Click the *Change User or Group* button. (This configures the task to run as `NTAUTHORITY\SYSTEM`.)
2. On the Triggers tab
   a. Set the event trigger or schedule to execute daily starting with the day the script will be deployed.
   b. Set the recurrence to every one (1) day and repeat every five (5) minutes for a duration of one (1) day.
3. On the Actions tab
   a. Set the task action to *Start a program*. The program name is `usbHistory.bat`.
   b. Set the script to start in the directory you designated earlier (see the above guidance on setting the `pth` variable).
4. In the Settings tab, check the box next to *Run task as soon as possible after a scheduled start is missed*.

Once you have configured the scheduled task and it begins to execute, when USB removable devices are inserted into the system, log files will begin to be generated in the designated directory. If you have configured `SYSLOG` correctly, the script will generate alerts as well.

2.5 OSSEC HIDS Integration

OSSEC is an open source, host-based intrusion detection system (HIDS) developed and maintained by Trend Micro [3]. OSSEC's architecture requires that you install individual agent daemons on each host in the network. The agents collect real-time information about their hosts and forward it to a central OSSEC server for analysis. You can configure the agents to monitor registry keys, log files, configuration file integrity, differences in the output of commands, and more. You can also configure the agent manually on each host or centrally through the OSSEC server. Since OSSEC is a fully featured system for host-based intrusion-detection and log analysis, we recommend that you take the time to understand the fundamentals of OSSEC's configuration and operation before implementing the tools described in this document. For more information about OSSEC, please refer to the *OSSEC Manual* [4].

2.5.1 Prerequisites

An OSSEC server should be present in your organization. During the testing of this control, we used a Fedora 14 virtual machine for the server installation; however, any popular server-class

[8] For more information about configuring scheduled tasks, refer to the Microsoft article titled "Configure a Scheduled Task Item" [16].

Linux distribution (Debian, Ubuntu, RHEL, etc.) should be adequate. In addition, you should install an OSSEC agent on each Windows client that you wish to monitor.

Please note that at the time of this writing, the latest release of OSSEC is version 2.6 (for both client and server). However, this version prevents the Windows agents from properly running command line tools when their configurations are deployed from the central server. Since our control depends on the agents' ability to monitor the output of command line tools, if organizations wish to deploy agents' configurations centrally, we recommend that they use version 2.5.1.

In addition to following these steps, you should ensure that you have installed the scripts in accordance with the steps outlined in Section 2.4.

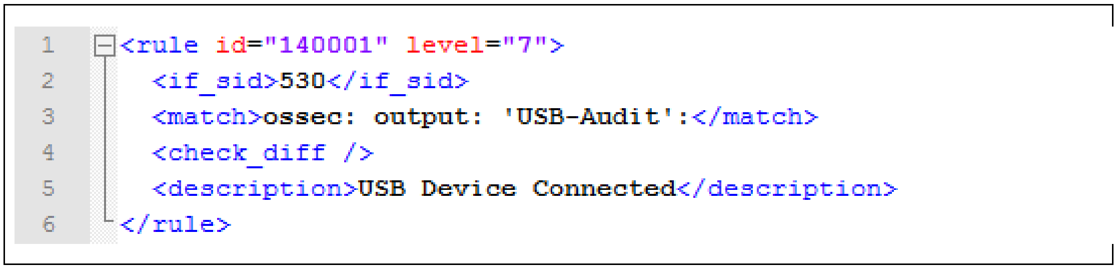
For more information about OSSEC, please refer to the *OSSEC Manual* [4].

2.5.2 OSSEC Configuration for USB Auditing

We utilized OSSEC in this report so that we could continually check the subkeys within the `USBSTOR` registry key to determine if a USB device has been inserted into a client system. We achieved this by configuring the agents to periodically run the open source `usbHistory.exe` command, which you must install on each client through either Group Policy or manual deployment.

When we used OSSEC to monitor the state of the `USBSTOR` registry, we encountered a problem: The agents' capabilities were not granular enough to monitor the specific values of the subkeys, such as timestamp and serial number information. However, OSSEC offers a useful feature that overcomes this limitation. Rather than monitor the presence of a registry key or file, OSSEC can monitor the output of a command line utility and detect changes in its output. Therefore, by configuring OSSEC to monitor the output of the `usbHistory.exe` tool—which prints out time stamp and serial number information—the host's OSSEC agent is now able to detect changes within the values of the subkeys and notify the OSSEC server.

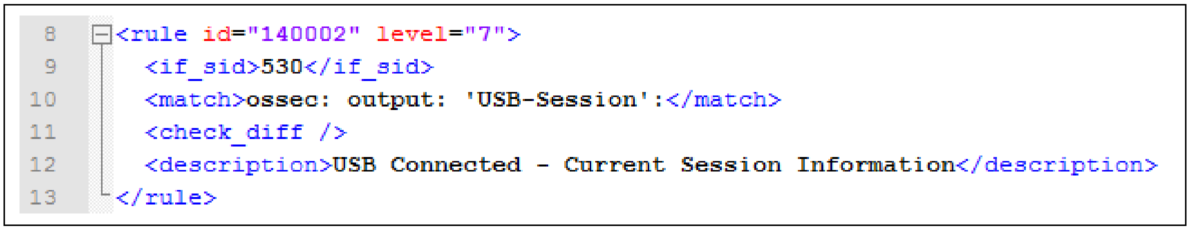
To do this, the OSSEC server must have two custom rules placed in its `local_rules.xml` file. In a default installation of OSSEC, this file can be found in the `/var/ossec/rules/` directory. The first rule (see Figure 5) configures the OSSEC server to log and create an alert anytime a USB device is inserted into the system.



```
<rule id="140001" level="7">
  <if_sid>530</if_sid>
  <match>ossec: output: 'USB-Audit':</match>
  <check_diff />
  <description>USB Device Connected</description>
</rule>
```

Figure 5: OSSEC Rule 140001

The second rule (see Figure 6) configures OSSEC to log the presence of any user sessions on the respective host where the USB device was connected. This allows you to take an “instant snapshot” of the host’s login sessions to help determine which user connected the USB device.

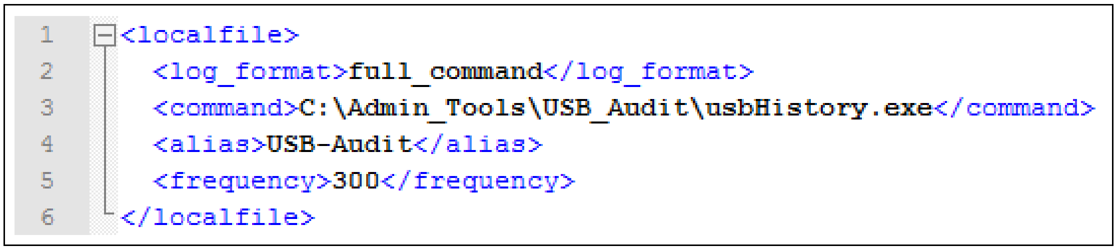


```
<rule id="140002" level="7">
  <if_sid>530</if_sid>
  <match>ossec: output: 'USB-Session':</match>
  <check_diff />
  <description>USB Connected - Current Session Information</description>
</rule>
```

Figure 6: OSSEC Rule 140002

The rules will be triggered only when logs arrive from the agent with a tag of *USB-Audit* or *USB-Session*, *and* the logs show differences from what was previously recorded. However, the agents must be configured to send these logs in the first place; you can do this either by manually editing the agents’ local `ossec.conf` file or through central deployment through the OSSEC server. For more information about how to centrally deploy agent configurations, see the document titled *Manual: Centralized Agent Configuration* [5].

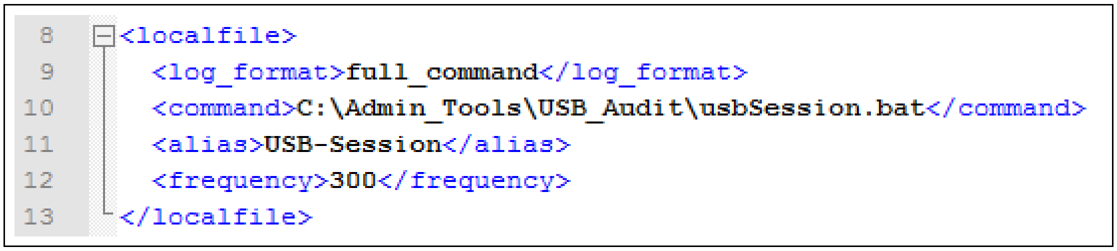
The agent configuration rule depicted in Figure 7 corresponds with the first rule (id=140001), depicted in Figure 5. This configuration rule instructs the OSSEC agent to run the `usbHistory.exe` command every 300 seconds, log the output, and tag it with `USB-Audit`. This tag instructs the server to match the log with rule 140001.



```
<localfile>
  <log_format>full_command</log_format>
  <command>C:\Admin_Tools\USB_Audit\usbHistory.exe</command>
  <alias>USB-Audit</alias>
  <frequency>300</frequency>
</localfile>
```

Figure 7: OSSEC Agent Configuration, Part 1

Similarly, the following agent configuration rule corresponds with rule 140002, depicted in Figure 6:



```
<localfile>
  <log_format>full_command</log_format>
  <command>C:\Admin_Tools\USB_Audit\usbSession.bat</command>
  <alias>USB-Session</alias>
  <frequency>300</frequency>
</localfile>
```

Figure 8: OSSEC Agent Configuration, Part 2

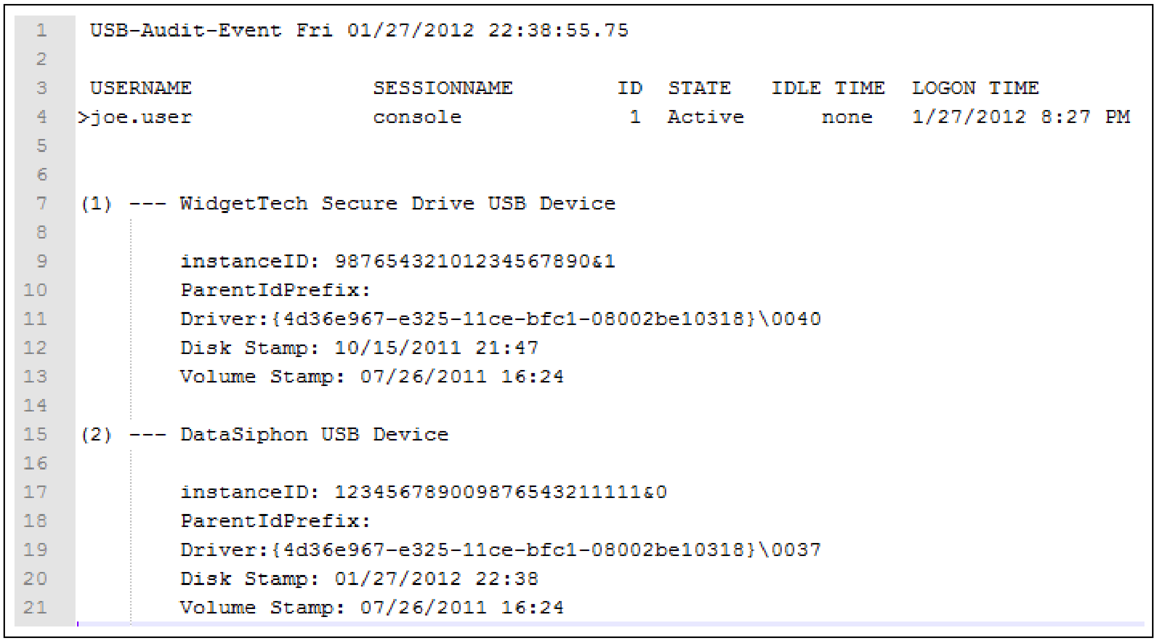
When the configuration is complete, make sure to restart the OSSEC server daemon as well as all client agent daemons. For environments with many hosts running OSSEC agents, those agents can be restarted centrally through the OSSEC server.

2.5.3 OSSEC Logs

By default, OSSEC stores the `diff` logs in the `/var/ossec/queue/diff` directory. A security analyst may easily become overwhelmed if removable devices are used on a regular basis throughout the organization. Therefore, a SIEM solution that is capable of monitoring directories of files will help administrators discern the information contained in these files. While Splunk is not typically referred to as a SIEM solution, it has SIEM-like capabilities that will allow IT staff to sift through the logs and uncover events of interest. We discuss Splunk further in Section 5.

3 Understanding USB Auditing Logs

When a user inserts a device into the system, the resulting USB event logs will contain information such as the username of the person who is currently logged in, the start time of his or her login session, and whether the session is currently active. In addition, each device that was ever connected to the system will be listed. If you conduct a search in Splunk for `USB-Audit-Event`, you will see records like the one depicted in Figure 9.



```
 1   USB-Audit-Event Fri 01/27/2012 22:38:55.75
 2
 3   USERNAME               SESSIONNAME        ID  STATE   IDLE TIME  LOGON TIME
 4  >joe.user               console             1  Active      none   1/27/2012 8:27 PM
 5
 6
 7  (1) --- WidgetTech Secure Drive USB Device
 8
 9          instanceID: 98765432101234567890&1
10          ParentIdPrefix:
11          Driver:{4d36e967-e325-11ce-bfc1-08002be10318}\0040
12          Disk Stamp: 10/15/2011 21:47
13          Volume Stamp: 07/26/2011 16:24
14
15  (2) --- DataSiphon USB Device
16
17          instanceID: 12345678900987654321111&0
18          ParentIdPrefix:
19          Driver:{4d36e967-e325-11ce-bfc1-08002be10318}\0037
20          Disk Stamp: 01/27/2012 22:38
21          Volume Stamp: 07/26/2011 16:24
```

Figure 9: USB Auditing Log

In Figure 9, line 1 shows the date and time the log was created. Line 4 shows the name of the user who is currently logged in; line 4 also shows the login date and time. Lines 7 and 15 list the manufacturer's name and model of the device. Lines 9 and 17 list the manufacturer's serial number for the device. Finally, lines 12 and 20 list the date and time when the device was last connected. Line 17 reflects the date and time that is closest to the log generation date and time reflected in line 1. This indicates that this is the device that spawned the audit event.

If we use this information combined with monitoring file-access audit logs around the time the device was inserted, we may get an indication that files were copied to the device. However, this does not *definitively* indicate that files were copied to the USB removable storage device.

In the example shown in Figure 9, two different device manufacturers are listed. If the organization only issues and permits the use of *WidgetTech* devices, then the *DataSiphon* device should raise concerns. And if a large number of sensitive file audit events are generated around the time an unauthorized USB storage device is inserted, this should generate a high-priority alert.

Note that if a user is logged into a machine but has momentarily stepped away after locking the console, USB audit events may still be generated by inserting a device into the locked computer.

When you conduct a Splunk query, you will be able to click various items in the search results. This allows IT personnel to explore additional patterns. For example, it is possible to click the device’s serial number to see other log events that contain the same serial number. This may allow IT personnel to identify whether the device was used on other systems in the organization.

4 Additional Benefits of Using USB Auditing Logs

The methods presented in this report guide organizations in their attempts to determine if a person may be of high risk for disclosing proprietary information, and one way of determining this is based on monitoring file access around the time of USB storage device utilization.

This section outlines additional benefits to government agencies and organizations working with intellectual property.

4.1 Federal Government Agencies

Organizations that use air-gapped systems, such as those that process classified information, may find it beneficial to use the tools presented in this report to search for devices with the same serial numbers on different systems. This would indicate a possible data spillage event.

***This approach would benefit federal government agencies significantly* since it allows them to detect when the same device is being used in systems where the classifications of those systems are different.**

4.2 Intellectual Property

A 2011 SEI report titled *Insider Threat Control: Using Centralized Logging to Detect Data Exfiltration Near Insider Termination* presents an example of an insider threat pattern based on the insight that “many insiders who stole their organization’s intellectual property stole at least some of it within 30 days of their termination” [6]. That report underscored the importance of developing a rule that could be applied to a log-indexing application to help analysts detect malicious behavior.

Organizations may wish to develop rules and associated alerts for the following conditions to better detect and respond to a possible malicious insider. The rules are unique to each auditing system or SIEM used and are therefore outside the scope of this report.

1. Attempted use of any non-company issued device.
2. Use of an unauthorized device around the same time as files containing intellectual property are accessed.
3. File accesses above what is considered normal for a particular role or job function.
4. File accesses above what is normal for a particular role or job function combined with using and analyzing USB removable media access logs.

A report containing the last 30 days of file accesses and USB removable media logs should be generated and reviewed by the information security team and/or management to identify any anomalies when an employee leaves the organization.

By implementing the above rules in a SIEM, this should reduce an analyst’s time reviewing logs by focusing on the high priority events generated by these rule sets.

5 Bringing it All Together with Splunk

5.1 Background

Splunk is a tool that helps IT administrators collect, index, monitor, and analyze machine-generated data using a web-style interface. This tool helps organizations

- troubleshoot application problems
- investigate security incidents easily and quickly
- efficiently identify data patterns
- provide metrics and diagnose problems

As such, Splunk can be used to quickly uncover events that may be of importance to an organization. In addition to using the USB auditing events described in this document, organizations should enable the auditing of sensitive files.[9] By sending this information to Splunk, organizations can quickly begin to form a full picture of what a data leakage incident looks like.

This report focuses on using a Splunk instance on Microsoft Windows Server 2008 R2 for managing event logs. Splunk has both a free and an enterprise version. One of the key differences in the free versus the enterprise version is the amount of logs that can be processed in one day; currently, the free version caps this at 500MB per day. Depending on the organization's needs, the free version may assist smaller organizations get started in logging. The enterprise version offers additional features and is capable of processing larger volumes of data [2]. We chose to focus on Splunk since a freely available version of it is available and it can quickly search large quantities of data. It is important to note that you will need to develop rules to monitor USB removable media devices since such rules are not part of Splunk's standard package. These rules will vary from organization to organization depending on the devices used and what will trigger an alert.

5.2 Actual Case

Consider the following actual case:

> The insider, presumably a foreign national, was employed by a foreign division of the victim organization, a construction and mining company. At the time of the incident, the insider was also working for a foreign internet technology organization. The insider used another employee's user ID and password to access the organization's server, which was located in the organization's headquarters in the United States. The insider downloaded over 4,000 confidential documents. Closed-circuit cameras captured visuals of the insider accessing the server at the time the files were downloaded. User logs also reflected the password and user ID the insider used to gain access to the server. Authorities discovered media—specifically a hard disk and a flash drive—where the insider stored the stolen files. The insider was arrested and convicted, but sentencing details are unavailable.

[9] A prior SEI report titled *Insider Threat Control: Understanding Data Loss Prevention (DLP) and Detection by Correlating Events from Multiple Sources* discusses how to audit sensitive files and folders [1].

In this case, the insider accessed over 4,000 confidential documents. If the organization had configured proper auditing, numerous auditing events would have been generated and sent to Splunk for analysis. By monitoring the Splunk dashboard or using an alert, an administrator would have noticed a spike in events over a short period of time. This could have indicated a problem that would have raised a red flag to signal that further investigation was necessary. By combining the USB device auditing events into Splunk, these events—a USB device event followed by a large number of file accesses by the same user—would have raised the priority of the alert. Furthermore, if the organization had issued a particular brand of flash drives or had maintained a list of known serial numbers for these devices, the insertion of an unrecognized USB device would have raised the alert level priority even further.

By implementing the methods described in this report, the organization would have been able to detect the incident earlier, thereby reducing the damage that the organization sustained.

5.3 Sending Information to Splunk

To easily and effectively analyze the logs generated by the tools presented in this document, the logs need to be fed into Splunk.[10] To collect the OSSEC logs, a Splunk server and a Splunk forwarding service on the OSSEC server are both required. It is possible to use the OSSEC server as a Splunk server as well, but we did not test this type of configuration. Depending on the quantity of logs and the server configuration, that approach may not advisable.

Follow these steps to feed the OSSEC logs into the Splunk server:

1. Install Splunk on a dedicated machine or on a virtual machine that is capable of processing the logs that will be sent to it.
2. Install Splunk on the OSSEC server as a forwarding agent and configure it to forward events to the Splunk server. You should configure the Splunk forwarder to monitor the `/var/ossec/queue/diff` directory on the OSSEC server. This will allow USB auditing events to be forwarded to the Splunk server. Be sure to configure the server to receive events from the forwarding client.[11] You may also need to open firewall ports on network firewalls and/or the hosts/servers.

Once you have configured everything properly, you should begin to see events within Splunk whenever a USB removable storage device is inserted into the system.

[10] Splunk is available for download on the Splunk website; the site also offers installation instructions [17].

[11] Instructions for configuring the server to receive events from the forwarding client are available on the Splunk website [17].

6 Conclusion

USB removable storage devices allow users to copy data from company systems. Some malicious insiders attempting to remove data from organizational systems may do so by using removable USB media. USB storage devices are becoming smaller and smaller while their storage capacities are becoming greater and greater. This presents a risk to the organization that must be addressed through policies and procedure and by technical means.

In this report, we presented methods that organizations can employ to increase awareness of which removable storage devices are used throughout the organization. While this is not a smoking gun that proves that data has been copied to a device, it does allow management to be aware of activities that are occurring in the organization. These methods also offer several key advantages that can assist an organization in its efforts to monitor potential incidents of data theft:

1. USB device usage can be detected quickly. It takes less than five minutes for the system to generate an alert.
2. There is redundant logging of information. Logs are stored locally on the hosts, centrally on an OSSEC server, and centrally on a SYSLOG server.
3. The system provides attribution. Current user-session information is logged when an incident is detected.
4. The system utilizes native and open source tools. Local logging utilizes only Windows native capabilities and a single, open source, forensic executable; centralized logging can be done with the open source OSSEC system.
5. The system is customizable. An organization can choose to exclude any of the centralized logging capabilities and retain just the local logging capabilities, or further modify the control to best suit its needs.

By implementing the auditing scripts and supporting tools described in this report, IT staff will be able to help interpret auditing logs, and this will make it easier to help management make informed decisions about possible threats to the organization.

Appendix: Scripts: `usbHistory.bat` and `usbSession.bat`

usbHistory.bat

Copy and paste the following code into a text file and save it into the script directory as `usbHistory.bat`.

```
@ECHO OFF
SETLOCAL
:: The below line tells the script where it is stored and where working files will be kept
:: Change the path below to reflect where it will be on the local machine.
SET pth="C:\Admin_Tools\USB_Audit"
%pth%\usbHistory.exe > %pth%\newlog.tmp

:: Check to see if the last.log file exists, if not, create it
IF NOT EXIST %pth%\last.log (
 TYPE NUL > %pth%\last.log
)

:: Compare the recent results with the last and save differences to output.tmp
FC %pth%\newlog.tmp %pth%\last.log > %pth%\output.tmp

:: Check to see if any changes in the system have occurred.
FINDSTR /m "no differences encountered" %pth%\output.tmp > NUL

:: The below lines apply formatting to the usb-audit.log file
IF %ERRORLEVEL% GTR 0 (
 IF EXIST %pth%\usb-audit.log (
  ECHO. >> %pth%\usb-audit.log
  ECHO. >> %pth%\usb-audit.log
  ECHO ------------------------------------ >> %pth%\usb-audit.log
 )

:: If the usb-audit.log file does not exist, create it
 IF NOT EXIST %pth%\usb-audit.log (
  TYPE NUL > %pth%\usb-audit.log
 )
:: The following lines capture system information and write it to various files
 ECHO A USB Device has recently been inserted into system: %COMPUTERNAME% > %pth%\message.log
 ECHO User Executing script: %USERDOMAIN%\%USERNAME% >> %pth%\usb-audit.log
 ECHO Computer: %COMPUTERNAME% >> %pth%\usb-audit.log
 ECHO Date/Time: %DATE% %TIME% >> %pth%\usb-audit.log
 ECHO. >> %pth%\usb-audit.log
 ECHO Session Information: >> %pth%\usb-audit.log
 ECHO. >> %pth%\usb-audit.log
:: The below line logs the current user session and their status to usb-audit.log
 CALL QUSER.EXE >> %pth%\usb-audit.log
 ECHO. >> %pth%\usb-audit.log
 ECHO New USBSTOR Registry Information: >> %pth%\usb-audit.log
 TYPE %pth%\newlog.tmp > %pth%\last.log
:: The lines below sanitize the output of usbHistory.exe and append the
:: output to usb-audit.log. Credit goes to nabiy for the usbHistory tool.
 TYPE %pth%\newlog.tmp | findstr /v "History Dump" | findstr /v nabiy > %pth%\newlog2.tmp
 TYPE %pth%\newlog2.tmp >> %pth%\usb-audit.log
:: Record date, time, and session information to session.log
:: These lines are needed for OSSEC to detect file changes
 ECHO %DATE% %TIME% > %pth%\session.log
 ECHO. >> %pth%\session.log
 CALL QUSER.EXE >> %pth%\session.log
 TYPE %pth%\newlog2.tmp >> %pth%\session.log
 DEL %pth%\newlog2.tmp
:: To DISABLE SYSLOG alerts, insert a double colon before the LOGPARSER command below
:: If you are using SYSLOG, change the IP address in the line below to that of your SYSLOG server
 LOGPARSER "SELECT * INTO @10.0.1.102:514 FROM '%pth%\message.log'" -i:TEXTLINE -o:SYSLOG -
protocol:UDP

:: Clean up temporary files
DEL %pth%\newlog.tmp
DEL %pth%\output.tmp

ENDLOCAL
```

usbSession.bat

Copy and paste the following code into a text file and save it into the script directory as `usbSession.bat`.

```
@ECHO OFF
SET pth="C:\Admin_Tools\USB_Audit"
TYPE %pth%\session.log
```

References

URLs are valid as of the publication date of this document.

[1] G. J. Silowash and C. King, "Insider Threat Control: Understanding Data Loss Prevention (DLP) and Detection by Correlating Events from Multiple Sources," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 2012.

[2] Splunk, Inc., "Free Vs. Enterprise," 2012. http://www.splunk.com/view/free-vs-enterprise/SP-CAAAE8W

[3] Trend Micro, "Welcome to the Home of OSSEC.” http://www.ossec.net/

[4] Trend Micro, "OSSEC Manual," 23 November 2009. http://www.ossec.net/main/manual

[5] Trend Micro, "Manual: Centralized Agent Configuration.” http://www.ossec.net/main/manual/centralized-config

[6] M. Hanley and J. Montelibano, "Insider Threat Control: Using Centralized Logging to Detect Data Exfiltration Near Insider Termination," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 2011.

[7] Splunk, Inc., "Free Vs. Enterprise," 2012. http://www.splunk.com/view/free-vs-enterprise/SP-CAAAE8W

[8] Microsoft, "Microsoft Technet," 2012. http://technet.microsoft.com/en-us/library/cc749183.aspx

[9] SourceForge.net, "USB History Dump." http://sourceforge.net/projects/usbhistory/

[10] Splunk Docs, "Installation Manual.” http://docs.splunk.com/Documentation/Splunk/latest/Installation/WhatsintheInstallationManual

[11] Splunk Docs, "Getting Data in Manual.” http://docs.splunk.com/Documentation/Splunk/latest/Data/Usingforwardingagents

[12] Nabiy, "USB History," SOURCEFORGE.NET. http://nabiy.sdf1.org/index.php?work=usbHistory

[13] Microsoft TechNet, "Configure a Scheduled Task Item.” http://technet.microsoft.com/en-us/library/cc725745.aspx

[14] Microsoft, "Log Parser 2.2.”
http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=24659

[15] Splunk, Inc., "What is Splunk?” http://www.splunk.com/product

[16] D. Melber, "Centralized Auditing Is Here and It's FREE!" 25 June 2009.
http://www.windowsecurity.com/articles/centralized-auditing-here-free.html

[17] Splunk, Inc., "Download Splunk. Install It in Minutes.”
http://www.splunk.com/download?r=/product

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE January 2013	3. REPORT TYPE AND DATES COVERED Final

4. TITLE AND SUBTITLE Insider Threat Control: Using Universal Serial Bus (USB) Device Auditing to Detect Possible Data Exfiltration by Malicious Insiders	5. FUNDING NUMBERS FA8721-05-C-0003
6. AUTHOR(S) George J. Silowash & Todd B. Lewellen	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2013-TN-003
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) ESC/CAA 20 Schilling Circle, Building 1305, 3rd Floor Hanscom AFB, MA 01731-2125	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES	
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS	12B DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Universal serial bus (USB) storage devices are useful for transferring information within an organization; however, they are a common threat vector through which data exfiltration can occur. Despite this, many organizations permit the use of USB devices on their systems. Implementing controls to track the use of these devices is necessary if organizations wish to retain situational awareness and auditing capabilities during a data theft incident.

This report presents methods to audit USB device use within a Microsoft Windows environment. Using various tools—the Windows Task Scheduler, batch scripts, Trend Micro's OSSEC host-based intrusion-detection system (HIDS), and the Splunk log analysis engine—we explore means by which information technology (IT) professionals can centrally log and monitor USB device use on Microsoft Windows hosts within an organization. In addition, we discuss how the central collection of audit logs can aid in determining whether sensitive data may have been copied from a system by a malicious insider.

14. SUBJECT TERMS Universal serial bus, USB, malicious activity, auditing, OSSEC	15. NUMBER OF PAGES 34
16. PRICE CODE	

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL